Graphics Silver Dollar Game

Due Sunday September 13, 2015

Objectives

For this assignment, you will:

- Refresh your memory of the Java programming language
- Gain practice using the `ArrayList` class
- Gain exposure to inner classes
- Gain exposure to Java Graphics
- Gain exposure to Java event handling

Description

In this assignment, you will create a graphical version of Bailey's Silver Dollar Game. Read Section 3.10 of the textbook *Java Structures* for a description of the game. In this version of the game, the user will use the mouse to move the coins instead of typing commands on the keyboard.

The correctness of the assignments in this class will be automatically verified. For this reason, you must follow all naming conventions specified in this assignment.

Classes

Coin

The `Coin` class represents a single coin. This class is already implemented for you.

CoinSquare

The `CoinSquare` class represents a square. This class is already implemented for you.

GraphicsCoinStrip

The `GraphicsCoinStrip` class uses the `Coin` and `CoinSquare` class to implement the Silver Dollar Game. This class is partially implemented. There are comments suggesting what you need to add. The `contains` methods that `Coin` and `CoinSquare` inherit from `Eclipse2D` and `Rectangle2D` may be helpful.

Notice that there are no `play` or `move` methods in the `GraphicsCoinStrip` class because the mouse is in control of the game. Much of what drives the game is the mouse event handling which can be found in the inner class `CoinMouseListener` inside the `GraphicsCoinStrip` class. The purpose of the inner class `CoinMouseListener` is to encapsulate all of the methods that deal with the mouse.

You can add whatever methods you think would be useful to the `GraphicsCoinStrip` or `CoinMouseListener` classes.

After you have a working copy of the game, write a method in this class that checks to see if the game is over and, if so, signal this to the user in some fashion. Possible examples are to print out a message to the console, or better, change the color of all of the coins. You can also make sure that the coins no longer move once the game is completed (although this is not required).

Getting Started

1. Read through the CS 062 style guide linked on the course web page under Documentation and Handouts. You must follow these guidelines for all of your assignments.
2. Create a new Java project in Eclipse named `Assignment01`. Add the `BAILEY` variable to the project. If you've forgotten how to do this, see the instructions on the Documentation page of the course website.
3. Using Finder, copy the starter files for this assignment into the `src` directory of your newly created project. The starter files can be found at `/common/cs/cs062/assignments/assignment01` inside of the `silverdollar` directory. Copy the entire `silverdollar` directory over into your `src` directory. ("silver dollar" is the package name.)
4. Refresh your project in Eclipse.
5. The `Coin` and `CoinSquare` classes are complete. You do not need to modify them. However, take a look at them to see what methods are available.
6. You are now ready to get started! This assignment asks you to fill in the constructor and add the appropriate methods in the `GraphicsCoinStrip` class to play the game. As much as possible, *try and develop incrementally. That is, get one small piece working and then move on to another piece.*
7. When you are done, read the Submitting Your Work section below.

Grading

You will be graded based on the following criteria:

criterion	points
game starts with random coin positions	1
coins can be dragged	1
coins can be dragged multiple squares	1
dropped coins end up centered in correct location	1
illegal moves are not allowed	3
game over is indicated correctly	2
general correctness	2
appropriate comments (including JavaDoc)	2
style and formatting	2
submitted correctly	1

NOTE: Code that does not compile will not be accepted! Make sure that your code compiles before submitting it.

Submitting Your Work

1. Before you submit, you must comment your code. We will be using the JavaDoc commenting style. To be compliant with JavaDoc, you *must* have the following:

  - Each comment should start with /** and end with **/. Every line in between should start with a * and be appropriately indented.
  - A comment describing the class right before the class declaration (i.e. after the `import` statements). This comment should include the `@author` tag after the class description, and the `@version` tag after the author tag.
  - A comment for each method describing what the method does. This comment should describe the *what* but not the *how*.
  - `@param`, `@return` and `@throws` tags for each method (when appropriate)
  - pre- and post- conditions as appropriate

2. Export your project from Eclipse

   - From within Eclipse, select “Export” from the “File menu.
   - Click on the triangle next to “General” in the dialog box. Select “File system” and click next.
   - Make sure all the files in the dialog on the right hand side are checked, and then click the “Browse” button next to the “To directory:” entry
   - Select “Desktop” from the pulldown menu and click on “Choose”.

3. Rename the exported directory.

   - Exporting your project from Eclipse will create a new folder on the Desktop. Rename this folder using the following convention: **Assignment01_LastNameFirstName**. For example, “Assignment01_BannisterMichael”

4. Drag your exported project folder to the dropbox

   - Now open the “cs062” folder on the desktop by double-clicking on it. Within the “cs062” folder you should see a “dropbox” folder.
   - Drag the folder you just created into the dropbox folder. When you do this, the computer may warn you that you will not be able to look at this folder. That is fine. Just click “OK”.
   - If you submit multiple times, change the name of the folder slightly. For example **Assignment01_LastNameFirstName_v2**. We will grade the latest submission made before the 11:59 p.m. deadline.